PRODUCT RELEASE

ObjectStore

ObjectStore Standard 2025.2

December 2025

IGNITETECH

ObjectStore

TABLE OF CONTENTS

CHANGE LOG

ObjectStore Standard 2025.2

NEW FEATURES

- **Minimum supported ping callback intervals reduced from 10 to 1 second.**

  Adjusted the minimum value for cache_mgr_ping_time and cache_mgr_ping_time_in_trans parameters from 10 seconds to 1 second.
  Any events or state changes that depend on these parameters will be detected and acted upon sooner, improving timeliness.

ObjectStore Standard 2025.1

NEW FEATURES

- **Support for Kerberos authentication on Windows**

  On Windows, ObjectStore now supports Kerberos as an authentication option. Starting with this version, NTLM (option NT_REMOTE) is no longer supported, as Microsoft is gradually removing it from Windows. For further information, see *"Managing ObjectStore"* guide: *Chapter 2*, section *"Authentication Required"*; *Chapter 3*, sections *"OS_AUTH"* and *"OS_SPN (Windows only)"*; *Chapter 4*, section "*osserver*"; and *Chapter 8*, section "*Use Active Directory for Authentication*".

- **Windows Server 2025 is now supported**

RESOLVED ISSUES

- **[GHI-12460] Fixes ""Error writing to (...)" when using OS_DEBUG_NETWORK=1"**

A fix was implemented to avoid race conditions that were responsible for causing log files to have "Error writing to (...)" in specific conditions when using the OS_DEBUG_NETWORK environment variable with value “1”.

- **[GHI-12565] Documentation updates regarding security configurations**

  A new subsection “*Configuring Process Permissions for Enhanced Security (Windows only)*” was added to the section “*Managing Processes*” of the Managing ObjectStore documentation file. This new subsection provides further details on how the environment variable OS_SECURITY_DESC_SDDL can be used to define fine-grained permission policies on the ObjectStore services via Security Descriptor Definition Language (SDDL).

- **[GHI-12262] Upgrade of libexpat library**

  Libexpat was upgraded to version 2.7.1.

ObjectStore Standard 2024.2

RESOLVED ISSUES

- **Documentation updates**

  Fixed version numbers in documentation and added missing environment variable settings.

- **[GHI-9586] Fix OOMresulting in uncontrolled crash**

  Specifically for the **GHI-9586** issue related to OOM during *osbackup* operation but should ensure clean shutdown in case of any OOM exception.

- **[GHI-10665] Resolve several CVE vulnerabilities**

  CVE-2021-36373: Processing specially crafted TAR archives can cause excessive memory allocation, leading to out-of-memory errors.

CVE-2021-36374: Similar vulnerability affecting ZIP and related archive formats, causing potential denial-of-service conditions.
CVE-2023-44487 (HTTP/2 Rapid Reset): A denial-of-service vulnerability in the HTTP/2 protocol allows attackers to flood servers with rapid reset frames, leading to resource exhaustion.
CVE-2022-2048: Jetty vulnerable where Invalid HTTP/2 requests can lead to denial of service.
CVE-2021-34429: URIs can be crafted using some encoded characters to access the content of the WEB-INF directory and/or bypass some security constraints.
CVE-2023-40167: Jetty accepts the '+' character proceeding the content-length value in a HTTP/1 header field. This is more permissive than allowed by the RFC and other servers routinely reject such requests with 400 responses.
CVE-2024-9823: DoSFilter leaks USER_AUTH entries.
CVE-2024-22201: 11259 HTTP/2 connection not closed after idle timeout when TCP congested.
CVE-2023-5072: Java: DoS Vulnerability in JSON-JAVA.
CVE-2022-45688: JSON stack overflow vulnerability.

ObjectStore Standard 2024.0

RESOLVED ISSUES

- **Incorrect file version for the executable binaries**

  File and product versions for DLLs and executables should now correctly be "2024.0".

- **[IBUENG-1924] Vulnerability in libexpat library**

  Libexpat was upgraded to version 2.6.2.

- **[IBUENG-1701] Compilation error when building with C++ 20**

  It should no longer break with the error *C2760: syntax error: 'register' was unexpected here; expected 'statement_seq'" when compiling*. Note that C++20 support is still a work in progress.

ObjectStore Standard 2023.1 (2013.0 update 25)

NEW FEATURES

- **Visual Studio 2022(versions up to 17.2) is now supported**

  All code that is linked with ObjectStore must be compiled with the **/d2FH4-** command line switch when 64-bit MSVC2019/MSVC2022 is used. ObjectStore does not support the FrameHandler4 format for exception handling. The **/Zc:implicitNoexcept-** and **/Zc:sizedDealloc-** command line switches are also mandatory.

- **Windows Server 2022 is now supported**

RESOLVED ISSUES

- **[OSCID-38179] ObjectStore assertion in objectstore::shutdown()**

  Assertion was removed as obsolete (it was used in 1999 to catch memory leak that was fixed in 2000).

- **[OSCID-38184] osverifydb bug**

  The bug in osverifydb was fixed. Osverifydb started to work quicker in fast mode if the database has a big number of bad pointers.

- **[OSCID-38154] Example dllschema lnk error on Windows**

  Example was fixed for MSVC2017.

NOTES

CERTIFIED PLATFORMS

Platform	Version

Java	8
Visual Studio	2017, 2019, 2022
Windows	2012 R2, 2016, 2019, 7, 8.1, 10, 11, 2022, 2025
GCC	4.4.7, 4.8
Red Hat Enterprise Linux	6.10, 7

KNOWN ISSUES

- New architectures were introduced in ObjectStore 2019.1 (2013.0 update 15). The Vector Header handling code was also changed.

  New architectures were added for MSVC2015 and MSVC2017, both x32 and x64. This leads to problems in interwork settings during the interaction between existing and newly added architectures when previous clients try to read data created or modified by the client with the new architecture. Existing clients could use architecture definition files to work around this issue.

  The error shown is like: *ObjectStore internal error <maint-0025-0131>Unknown architecture code 53. (err_internal)*

  Vector Header handling code was changed for MSVC2015 and MSVC2017 x64 platforms due to the changes coming from a bugfix by Microsoft in its x64 compiler code. Unfortunately, this parameter is not alterable via architectural definition files because the behavior is complicated. Therefore it is worked around by setting an environment variable OS_FIX_VECTOR_HEADERS to 1 or solved by upgrading ObjectStore Client installations to an ObjectStore 2019.1 (2013.0 update 15) or later.

  The error shown is like: *An incorrect vector header was found during inbound relocation at 0X000000003000 3D80. The source architecture is an Intel platform with 64-bit Windows and Visual C++ 141 or higher. The address corresponds to offset 0x3d80 within cluster # 0, segment #0 of database DATABASE_NAME.*

  There is a **workaround** for these issues:

  1. Providing an architecture definition file for use with existing client installations. The architecture definition file is shipped in the release package.

2. Setting the environment variable OS_FIX_VECTOR_HEADERS to 1 - this makes sure that vector headers created by MSVC2015/2017 x64 could be read by the existing clients.

There is a **solution** to these issues:

1. Rebase the existing ObjectStore Client installations to ObjectStore Standard 2019.1 (2013.0 update 15) or later for the corresponding platform.

Those issues affect only the ObjectStore Client; the ObjectStore Server and the Cache manager are unaffected, so any existing ObjectStore Server installation could continue to be used without any modifications.

- New architectures were introduced in ObjectStore 2020.1 (2013.0 update 22).

  New architectures were added for MSVC2019 x32 and x64.

  The Vector Header handling code was NOT changed. MSVC2019 x64 vector headers have the same format as MSVC2015 and MSVC2017.

- New architectures were introduced in ObjectStore 2023.1 (2013.0 update 25).

  New architectures were added for MSVC2022 x32 and x64.

  The Vector Header handling code was NOT changed. MSVC2022 x64 vector headers have the same format as MSVC2015, MSVC2017, MSVC2019.

- Databases growing above 1 TB can cause server crashes and data loss.

  The ObjectStore server can't handle the databases reaching its theoretical size limitations correctly. The servers which reach these constraints will crash, and this crash will likely result in the loss or corruption of the latest data submitted to it.

  For the file-based databases, the file size can be monitored, and when this size approaches 1 TB it should be compacted, or separated into multiple databases.

Note that the 1 TB limit is related to the internal ObjectStore structures - it's possible that the server reaches this database size way before the logical database size reaches this threshold. If the stored data is significantly smaller, the oscompact utility should be used to reduce the physical database size.

- The indexed cursor iteration order might be not correct when multiple virtual inheritance is used.

  When the member that is the subject of indexing is in one of the virtually inherited structs the sorting function fails. The workaround for this issue is to move the member being indexed and sorted on to the leaf node.

- C++11 and higher (Modern C++) features are not supported in ObjectStore contexts.

  Microsoft Visual Studio 2015 introduced advanced support of C++11/C++14 standard, and Microsoft Visual Studio 2017 improved the standard compliance further. All newly introduced features (e.g. lambdas) are not supported in ObjectStore contexts - e.g. in ObjectStore transactions, TIX exception capture blocks, etc. It is still possible to use Modern C++ in other parts of the application.

- /Zc:implicitNoexcept- /Zc:sizedDealloc- compiler command-line switches are mandatory for Visual Studio 2015 and later.

  Microsoft Visual Studio 2015 introduced the support for the C++11 standard.

  The C++11 standard includes the noexcept specifier, which is implicit for destructors. ObjectStore does throw from destructors, so using the /Zc:implicitNoexcept- compiler command-line switch is mandatory for the compilation of the code that is linked to ObjectStore.

  The C++11 standard includes sized deallocation. ObjectStore defines overloads of new and delete operators that conflict with sized deallocation, so using the /Zc:sizedDealloc- compiler command-line switch is mandatory for the compilation of the code that is linked to ObjectStore.

HOW TO EMBED OBJECTSTORE COMPONENTS INTO ANOTHER INSTALLER

1. Install the required components of ObjectStore on the development machine.
2. Zip up all files in the installation directory.
3. Bundle the zip into your installer package.
4. Add instructions into the installer to:
    4.1. Extract the bundled zip
    4.2. Create below registry settings.
    4.3. Start the ObjectStore services.

Here is the list of registry entries and system environment variables that should be applied to the target machine. They are shared by the ObjectStore components and named according to the legacy installer. The "${OS_INSTALL_DIR}" should be replaced in runtime to the path of unzipped ObjectStore components.

C++ Interface - DBMS Client and Server

```
setx OS_ROOTDIR = ${OS_INSTALL_DIR}\OStore
setx OS_TMPDIR = ${OS_INSTALL_DIR}\temp
setx LIB = ${OS_INSTALL_DIR}\OStore\lib;%LIB%
setx INCLUDE = ${OS_INSTALL_DIR}\OStore\include;%INCLUDE%
setx PATH = ${OS_INSTALL_DIR}\OStore\bin;%PATH%

[HKEY_LOCAL_MACHINE\SOFTWARE\ObjectStore Inc.\ObjectStore 2013\Registration\]
"ObjectStore ExamplesDirectory"=""
"ObjectStore HTML Documentation Directory"=""
"ObjectStore Install Directory"="${OS_INSTALL_DIR}\OStore"
"ObjectStore PDF Book Files Directory"=""
"ObjectStore Single Directory"=""
"ObjectStore Suite Top Directory"=${OS_INSTALL_DIR}"
"Program Folder Name"="ObjectStore Win64 2013.0"
"Update"="0"

[HKEY_LOCAL_MACHINE\SOFTWARE\ObjectStore Inc.\ObjectStore 2013\Server\]
"Auto Start Server"="1"

[HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\ObjectStore Cache Manager R7.0\]
"DelayedAutostart"=dword:00000001
"DisplayName"="ObjectStore Cache Manager R7.0"
"ErrorControl"=dword:00000001
```

```
"ImagePath"="${OS_INSTALL_DIR}\OStore\BIN\OSCMGR6.EXE"
"ObjectName"="LocalSystem"
"Start"=dword:00000002
"Type"=dword:00000010
"WOW64"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\ObjectStore Server R7.0\]
"DelayedAutostart"=dword:00000001
"DisplayName"="ObjectStore Server R7.0"
"ErrorControl"=dword:00000001
"ImagePath"="${OS_INSTALL_DIR}\OStore\BIN\OSSERVER.EXE"
"ObjectName"="LocalSystem"
"Start"=dword:00000002
"Type"=dword:00000010
"WOW64"=dword:00000001
```

DSA as Server

Same as in C++ Interface plus:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\dsa\]
"Description"="ObjectStore System Administration and Monitoring"
"DisplayName"="Data Services Administrator"
"ErrorControl"=dword:00000001
"ImagePath"="${OS_INSTALL_DIR}\sysadmin\bin\wrapper.exe -s
${OS_INSTALL_DIR}\sysadmin\bin\conf\wrapper.conf"
"ObjectName"="LocalSystem"
"Start"=dword:00000002
"Type"=dword:00000010

[HKEY_LOCAL_MACHINE\SOFTWARE\ObjectStore Inc.\ObjectStore 2013\DataServices\]
"UseType"="DSA Server"
```

DSA as Process Manager

Same as in DSA as Server except:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ObjectStore Inc.\ObjectStore 2013\DataServices\]
"UseType"="DSA Process Manager"
```

DDML

Same as in C++ Interface except:

```
setx PATH = ${OS_INSTALL_DIR}\DDML\bin;${OS_INSTALL_DIR}\OStore\bin;%PATH%
```

Java Interface

```
setx OS_ROOTDIR = ${OS_INSTALL_DIR}\OStore
setx OS_TMPDIR = ${OS_INSTALL_DIR}\temp
setx LIB = ${OS_INSTALL_DIR}\OStore\lib;%LIB%
setx INCLUDE = ${OS_INSTALL_DIR}\OStore\include;%INCLUDE%
setx PATH = ${OS_INSTALL_DIR}\OSJI\bin;${OS_INSTALL_DIR}\OStore\bin;%PATH%
setx OSJI_ROOTDIR = ${OS_INSTALL_DIR}\OSJI
```

JMTL

Same as in Java Interface except:

```
setx PATH =
${OS_INSTALL_DIR}\JMTL\bin;${OS_INSTALL_DIR}\OSJI\bin;${OS_INSTALL_DIR}\OStore\bin;%PATH%
```

IGNITETECH UNLIMITED

If you would like to upgrade your support plan to take advantage of the **Unlimited Program features**, please contact us for more information. If you have any questions about this release, please open a **support ticket** and our support team will be happy to assist you.

To ensure all of the appropriate staff within your organization are informed about important product updates, please notify **success@ignitetech.com** with emails for individuals who should receive these announcements.